

日 本 国 特 許 庁
JAPAN PATENT OFFICE

PCT/JP 03/05319

25.04.03

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2002年 4月26日

出 願 番 号

Application Number:

特願2002-125838

[ST.10/C]:

[JP2002-125838]

出 願 人

Applicant(s):

学校法人日本大学

REC'D 20 JUN 2003

WIPO

PCT

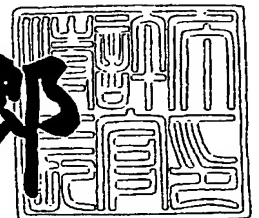
BEST AVAILABLE COPY

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH
RULE 17.1(a) OR (b)

2003年 6月 2日

特許庁長官
Commissioner,
Japan Patent Office

太田信一郎



出証番号 出証特2003-3041503

【書類名】 特許願

【整理番号】 ND0101P004

【あて先】 特許庁長官殿

【発明者】

 【住所又は居所】 東京都千代田区九段南四丁目 8 番 2 4 号 学校法人 日
 本大学内

 【氏名】 山本 登

【特許出願人】

 【識別番号】 899000057

 【住所又は居所】 東京都千代田区九段南四丁目 8 番 2 4 号

 【氏名又は名称】 学校法人 日本大学

【代理人】

 【識別番号】 100107113

 【弁理士】

 【氏名又は名称】 大木 健一

【手数料の表示】

 【予納台帳番号】 082590

 【納付金額】 21,000円

【提出物件の目録】

 【物件名】 明細書 1

 【物件名】 図面 1

 【物件名】 要約書 1

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 並列マージソート処理装置及び方法並びにプログラム

【特許請求の範囲】

【請求項 1】 複数のプロセッサを含み、入力データに対してマージソート処理を行う並列プロセッサと、マージソート処理の過程における中間のマージ結果であるデータ列を受け、これを所定の分割条件のもとで要求された数に分割し少なくともその一部を組にして前記並列プロセッサに返す領域分割部とを備え、前記並列プロセッサは、前記領域分割部から受けた複数の部分データ列の組を前記複数のプロセッサにそれぞれ割り当てて並列にマージ処理を行わせることを特徴とする並列マージソート処理装置。

【請求項 2】 部分データ列の組を割り当てられた前記プロセッサはそれぞれマージ処理を行うとともに当該マージ処理の結果を予め定められた領域の部分にそれぞれ出力し、前記領域の全体がマージ結果あるいは中間のマージ結果として出力されることを特徴とする請求項 1 記載の並列マージソート処理装置。

【請求項 3】 前記部分データ列の組それぞれに対して 2 つのプロセッサが割り当てられ、第 1 のプロセッサは前記部分データ列のキー値の大きい側の端から正順マージ処理を行い、第 2 のプロセッサは同じ部分データ列のキー値の小さい側の端から逆順マージ処理を同時に行い、前記第 1 のプロセッサは前記領域の部分のキー値の大きい側の端から順にマージ処理の結果を書き込み、前記第 2 のプロセッサは前記領域の部分のキー値の小さい側の端から順にマージ処理の結果を書き込むことを特徴とする請求項 2 記載の並列マージソート処理装置。

【請求項 4】 前記分割条件は、予めソートされた第 1 データ列 A が 2 つの部分データ列 A 1 と部分データ列 A 2 あるいはそれ以上に分割され、予めソートされた第 2 データ列 B が 2 つの部分データ列 B 1 と部分データ列 B 2 あるいはそれ以上に分割される場合において、少なくとも（部分データ列 A 1 の末尾のデータのキー値又は部分データ列 B 1 の末尾のデータのキー値のうちの最小値又は小さい方） \geq （部分データ列 A 2 の頭部のデータのキー値又は部分データ列 B 2 の頭部のデータのキー値のうちの最大値又は大きい方）が成立することであることを特徴とする請求項 1 記載の並列マージソート処理装置。

【請求項 5】 複数のプロセッサを含む並列プロセッサを用いて並列マージソート処理を行うための方法であって、

処理対象データを分割して複数の部分データ列を得る第 1 ステップと、

前記複数の部分データ列にそれぞれプロセッサを割り当てる第 2 ステップと、

割り当てられたプロセッサにより前記複数の部分データ列についてそれぞれ独立にソート処理を行う第 3 ステップと、

部分データ列を所定の分割条件のもとでそれぞれ要求された数に分割する第 4 ステップと、

分割された部分データ列のうちの少なくとも一部を組にする第 5 ステップと、

得られた組に対してプロセッサを割り当てる第 6 ステップと、

割り当てられたプロセッサによりマージ処理を行う第 7 ステップと、

マージ処理されたデータ列を前記部分データ列として前記第 4 ステップ乃至前記第 7 ステップを繰り返すステップと、を備える並列マージソート処理方法。

【請求項 6】 前記第 6 ステップにおいて、前記部分データ列の組に対して 2 つのプロセッサが割り当てられ、

前記第 7 ステップにおいて、第 1 のプロセッサは前記部分データ列のキー値の大きい側の端から正順マージ処理を行い、第 2 のプロセッサは同じ部分データ列のキー値の小さい側の端から逆順マージ処理を同時に行う、ことを特徴とする請求項 5 記載の並列マージソート処理方法。

【請求項 7】 前記第 4 ステップの分割条件は、予めソートされた第 1 データ列 A が 2 つの部分データ列 A 1 と部分データ列 A 2 あるいはそれ以上に分割され、予めソートされた第 2 データ列 B が 2 つの部分データ列 B 1 と部分データ列 B 2 あるいはそれ以上に分割される場合において、少なくとも（部分データ列 A 1 の末尾のデータのキー値又は部分データ列 B 1 の末尾のデータのキー値のうちの最小値又は小さい方） \geq （部分データ列 A 2 の頭部のデータのキー値又は部分データ列 B 2 の頭部のデータのキー値のうちの最大値又は大きい方）が成立することであることを特徴とする請求項 5 記載の並列マージソート処理方法。

【請求項 8】 前記第 4 ステップは、

それぞれ降順に整列された n 個のデータ列を含む 2 つの領域 D_1 と D_2 が与えられた

場合において、分割により得られた領域 D_1 の部分データ列と領域 D_2 の部分データ列の組み合わせの各組に含まれる最小のキー値をもつデータが D_1 域と D_2 域の先頭から 1, 2, 3 と数えて $2(n/k)$ 番目、 $4(n/k)$ 番目、 $6(n/k)$ 番目、 \dots 、 $2(k-1)(n/k)$ 番目のいずれかとなるように分割するために、

指標値（領域の先頭のデータを 0 とし、順次 1, 2, \dots となる）が (n/k) 、 $2(n/k)$ 、 $3(n/k)$ 、 \dots 、 $(k-1)(n/k)$ であるデータの少なくともいずれかを境界として選ぶステップと、

前記領域 D_1 と D_2 の作業用指標 i と j に境界の指標値を設定するステップと、

前記領域 D_1 の指標 i をもつデータのキー値と前記領域 D_2 の指標 j をもつデータのキー値を比較する比較ステップと、

最初の比較で前記領域 D_1 の指標 i をもつデータのキー値と前記領域 D_2 の指標 j をもつデータのキー値が等しくない場合、キー値の大きい方の指標値に 1 を加え、小さい方の指標値から 1 を引いた後、前記比較ステップに分岐するステップと、

2回目以降の比較で前記領域 D_1 の指標 i をもつデータのキー値と前記領域 D_2 の指標 j をもつデータのキー値との間の大小関係が前回の大小関係と逆転しない場合、大きい方の指標値に 1 を加え、小さい方の指標値から 1 を引いた後、前記比較ステップに分岐するステップと、

前記領域 D_1 の指標 i をもつデータのキー値と前記領域 D_2 の指標 j をもつデータのキー値が等しいとき、指標 i のキー値と指標 j のキー値をそれぞれ分割の境界とするステップと、

前記領域 D_1 の指標 i のキー値と前記領域 D_2 の指標 j のキー値との間の大小関係が前回の大小関係と逆転したとき、前回の比較操作における前記領域 D_1 のキー値と前記領域 D_2 のキー値の内の大きい方と、今回の比較操作における前記領域 D_1 のキー値と前記領域 D_2 のキー値の内の大きい方を比較して、小さい方の数値を分割の境界とするとともに、当該データの当初の比較相手を他方の領域の境界とするステップと、を備えることを特徴とする請求項 5 記載の並列マージソート処理方法。

【請求項 9】 前記第 4 ステップは、

それぞれ降順に整列された n 個のデータ列を含む2つの領域 D_1 と D_2 が与えられた場合において、領域 D_1 と D_2 をそれぞれ分割により得られた領域 D_1 の部分データ列と領域 D_2 の部分データ列を組み合わせた各組に含まれる最小のキー値をもつデータが D_1 域と D_2 域の先頭から1, 2, 3と数えて $2(n/k)$ 番目、 $4(n/k)$ 番目、 $6(n/k)$ 番目、 \dots 、 $2(k-1)(n/k)$ 番目のいずれかとなるように分割するために、

指標値（領域の先頭のデータを0とし、順次1, 2, \dots となる）が $(n/k)-1$, $2(n/k)-1$, $3(n/k)-1$, \dots , $(k-1)(n/k)-1$ であるデータの少なくともいずれかを境界として選ぶステップと、

前記領域 D_1 と D_2 の作業用指標 i と j に境界の指標値を設定するステップと、

前記領域 D_1 の指標 i をもつデータのキー値と前記領域 D_2 の指標 j をもつデータのキー値を比較する比較ステップと、

最初の比較で前記領域 D_1 の指標 i をもつデータのキー値と前記領域 D_2 の指標 j をもつデータのキー値が等しくない場合、キー値の大きい方の指標値に1を加え、小さい方の指標値から1を引いた後、前記比較ステップに分岐するステップと、

2回目以降の比較で前記領域 D_1 の指標 i をもつデータのキー値と前記領域 D_2 の指標 j をもつデータのキー値との間の大小関係が前回の大小関係と逆転しない場合、大きい方の指標値に1を加え、小さい方の指標値から1を引いた後、前記比較ステップに分岐するステップと、

前記領域 D_1 の指標 i をもつデータのキー値と前記領域 D_2 の指標 j をもつデータのキー値が等しいとき、指標 i をもつデータのキー値と指標 j をもつデータのキー値をそれぞれ分割の境界とするステップと、

前記領域 D_1 の指標 i をもつデータのキー値と前記領域 D_2 の指標 j をもつデータのキー値との間の大小関係が前回の大小関係と逆転したとき、前回の比較操作における前記領域 D_1 のキー値と前記領域 D_2 のキー値の内の小さい方と、今回の比較操作における前記領域 D_1 のキー値と前記領域 D_2 のキー値の内の小さい方を比較して、大きい方を分割の境界とするとともに、当該データの比較相手を他方の領域の境界とするステップと、を備えることを特徴とする請求項5記載の並列マージソート処理方法。

【請求項 1 0】 2 組のソート済のデータを任意の並列度のマージソートで整列させる並列マージソート処理方法であって、

大きさの順に整列された複数のキー値をそれぞれ含む第 1 のデータ列及び第 2 のデータ列を用意するステップと、

要求される並列度に応じて前記第 1 のデータ列及び前記第 2 のデータ列の部分列の組を作り、組単位にマージ処理をしても全体としてキーの並びに矛盾がないようにそれぞれ分割するステップと、

分割されたデータ列間で並列マージを行うステップとを備える並列マージソート処理方法。

【請求項 1 1】 複数のプロセッサを含む並列プロセッサを用いて並列マージソート処理を行うためのプログラムであって、

処理対象データを分割して複数の部分データ列を得る第 1 ステップと、

前記複数の部分データ列にそれぞれプロセッサを割り当てる第 2 ステップと、

割り当てられたプロセッサにより前記複数の部分データ列についてそれぞれ独立にソート処理を行う第 3 ステップと、

第 3 ステップ又は第 4 ステップで得た部分データ列を所定の分割条件のもとでそれぞれ要求された数に分割する第 4 ステップと、

分割された部分データ列のうちの少なくとも一部を組にする第 5 ステップと、

得られた組に対してプロセッサを割り当てる第 6 ステップと、

割り当てられたプロセッサによりマージ処理を行う第 7 ステップと、

マージ処理されたデータ列を前記部分データ列として前記第 4 ステップ乃至前記第 7 ステップを繰り返すステップと、を実行させるためのプログラム。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

この発明は、入力されたデータに対して並列プロセッサを用いて整列（ソート）操作を行うとともに併合（マージ）操作を行い、昇順又は降順に整列された 1 系列のデータ列を得るための並列マージソート処理装置及び方法並びにプログラムに関する。

【0 0 0 2】

【従来の技術】

コンピュータを用いた処理のひとつとして整列（ソート）処理及び併合（マージ）処理が知られている。これらの処理は、入力されたデータを昇順又は降順に整列させて1系列のデータ列を得るものである。併合（マージ）操作は、昇順（または降順）に整列（ソート）された2系列以上のデータ（整列対象のデータまたはレコードの総称）列を受けて、昇順または降順に整列された1系列のデータ列を出力するものである。

【0 0 0 3】

入力系列の数は、2、3、4・・・と種々の場合が考えられるが、特に2の場合は1回の比較で1単位 of データが整列されるため効率が良い。

【0 0 0 4】

併合（マージ）処理に係る所要時間のオーダーはデータ数(n)に比例するため、 n^2 または $n \log_2 n$ に比例する整列操作に比べ有利である。しかしデータ数の増大と共に所要時間も増すので、より高速なアルゴリズム（算法）が求められている。また性能の高いアルゴリズムの開発により、マージソートの適用範囲がさらに広まる可能性もある。だが2つのキー値の比較という単純な操作であるため、時間短縮の要因が潜在する可能性は低い。

【0 0 0 5】

並列プロセッサを用いて多数のデータに対して同時にマージソート処理を行い、処理時間を短縮することが提案されている。並列二分木マージソート算法もそのひとつであり、図13はその概略フローチャートであり、図14はプロセッサ数=8の場合の処理の説明図である。

【0 0 0 6】

図13を参照すると、処理対象データ（入力データ）をプロセッサの数に対応する k 個の区画（ただし、 $k = 2^n$ 、 n は整数）に分割する（S100）。 k 個のプロセッサを用いて分割された k 区画それぞれを、例えばクイックソート法を用いて独立にソートする（S101）。入力データに対して同時並行的に処理可能であり、1台のプロセッサの場合に比べて処理時間が短縮される。ソートされ

た k 区画のデータ列に対して n 段のマージソートを繰り返す (S 1 0 2) ことにより、最終的に全体として整列された 1 組のデータ列を得ることができる (S 1 0 3)。

【 0 0 0 7 】

図 1 4 を参照してプロセッサの数が 8 の場合の処理について説明する。図中、丸はプロセッサを示し、四角はデータ列あるいはデータ列 (D) が記憶される領域を示す。丸の中の記号はプロセッサが行う処理の内容を示し、S は整列 (ソート) 処理、M は併合 (マージ) 処理、V は必要に応じてなされる他の記憶領域への転送処理を意味する。

【 0 0 0 8 】

入力領域に置かれたデータは 8 つの部分に分割される。8 つのプロセッサ P 1 ~ P 8 によりこれら部分について同時並行的に整列処理を行う。これらの処理結果は領域 D 1 1 ~ D 1 8 に置かれる。

【 0 0 0 9 】

次に、第 1 段階のマージソートを行う。プロセッサ P 1 によりデータ列 D 1 1 と D 1 2 が併合されて領域 D 2 1 に置かれ、プロセッサ P 3 によりデータ列 D 1 3 と D 1 4 が併合されて領域 D 2 2 に置かれ、プロセッサ P 5 によりデータ列 D 1 5 と D 1 6 が併合されて領域 D 2 3 に置かれ、プロセッサ P 7 によりデータ列 D 1 7 と D 1 8 が併合されて領域 D 2 4 に置かれる。なお、第 1 段階においてプロセッサ P 2, P 4, P 6, P 8 は使用されない (プロセッサの割り当ては便宜上のものである。以下同じ)。

【 0 0 1 0 】

次に、第 2 段階のマージソートを行う。プロセッサ P 1 によりデータ列 D 2 1 と D 2 2 が併合されて領域 D 3 1 に置かれ、プロセッサ P 5 によりデータ列 D 2 3 と D 2 4 が併合されて領域 D 3 2 に置かれる。なお、第 2 段階においてプロセッサ P 2, P 3, P 4, P 6, P 7, P 8 は使用されない。

【 0 0 1 1 】

次に、第 3 段階のマージソートを行う。プロセッサ P 1 によりデータ列 D 3 1 と D 3 2 が併合されて領域 D 4 に置かれる。これでマージソートが完了する。な

お、第3段階においてプロセッサP2, P3, P4, P5, P6, P7, P8は使用されない。得られた結果D4は8つのプロセッサにより出力領域に転送される。

【0012】

【発明が解決しようとする課題】

上述した従来の手法によればマージ処理の過程において使用されなくなるプロセッサが存在し、プロセッサの処理能力が無駄になっていた。上記の例では第1段階の処理においてプロセッサの50%しか使用されず、第2段階の処理においてプロセッサの使用率は25%、第3段階の処理においてプロセッサの使用率は12.5%と、処理が進むにつれて使用されなくなるプロセッサが増えていく。これはマージ処理の過程においてデータ列の数が順次減少するので、プロセッサの数がデータ群の数よりも多くなるためである。

また、1台のプロセッサでマージするデータの数も後の段階になるごとに倍増していき、処理時間が増大する。プロセッサの使用率が低下する以上に1台のプロセッサが担当するデータ量が増大し、その結果、処理時間が増大してしまうことが問題である。これではせっかく並列プロセッサを利用しても処理はあまり効率化されず、処理時間の短縮も望めない。

【0013】

この発明に係る課題を解決するためになされたもので、並列プロセッサを用いたマージソート処理においてプロセッサの使用効率を高め、処理時間を短縮可能な並列マージソート処理装置及び方法並びにプログラムを提供することを目的とする。

【0014】

【課題を解決するための手段】

本発明は、複数のプロセッサを並列に接続してなる並列計算機において、各プロセッサの使用効率を高めることにより処理速度を向上させるものである。並列計算機を用いて行う処理のひとつであるソート処理は、入力データを分割してそれぞれについてソート処理を行うとともに、得られた複数のソート結果をマージしてひとつにまとめることにより最終的な処理結果を得るものである。本発明に

よればマージすべきデータ（図 1 4 に示したソートやマージによる部分的な整列結果）を分割することにより分割数の 2 倍の数のプロセッサを使用することができ、その結果、整列に要する処理時間を短縮できる。

【 0 0 1 5 】

また、プロセッサの使用効率を高める手法として次のことも考えられる。ひとつのデータ列に対して 2 つのプロセッサを用いて、キー値の大きい側（始端）から併合操作を始めて結果を出力域の始端から降順に出力する方法と、キー値の小さい側（終端）から併合操作を始めて出力域の終端から昇順に出力する方法の両方のマージ操作を同時平行的に実行するのである。この方式自体に起因する制約はない。このため、複数のタスクが並行に動作できる環境下では容易に実現できる。ここで、前者を正順併合、後者を逆順併合と呼ぶことにする。この方法によれば並列度は 2 倍に向上し、処理時間も半減する可能性が高い。しかし 2 倍を超える性能改善はこの方式のみでは十分でなく、上記の手法との併用が望ましい。

【 0 0 1 6 】

以下の説明において、説明の便宜上『一つのキーによる降順の整列』に限定する。整列に関連する用語には原則として『整列』と『併合』を用いるが、同義語的に『ソート』や『マージ』を用いることがある。

【 0 0 1 7 】

本発明に係る並列マージソート処理装置は、従来方式のマージ操作における 2 系列の入力データ列の組を複数組の 2 系列の入力データ列に分割する機構と、それらの組に対して並列にマージ操作を行う機構を基本要素とする。両機構をどのようにプロセッサ群に割り付けるかは任意である。すなわち、領域分割の仕事を複数のプロセッサにさせてから、マージ操作ではプロセッサを割り当て直して実行させることもできるし、領域分割とその後のマージ操作を同一プロセッサに担当させることもできる。

それぞれのマージ操作の出力結果は、あたかも 1 つのマージ操作を行った場合と同じ結果でなければならない。このため、並行に実行される各マージ操作では分割した入力データ列の組の全体の順序関係に対応した出力域に結果を出力しなければならない。

【 0 0 1 8 】

部分データ列の組を割り当てられた前記プロセッサはそれぞれマージ処理を行うとともに当該マージ処理の結果を予め定められた領域の部分にそれぞれ出力し、前記領域の全体がマージ結果あるいは中間のマージ結果として出力される。

【 0 0 1 9 】

前記部分データ列の組それぞれに対して2つのプロセッサが割り当てられ、第1のプロセッサは前記部分データ列の一方の端（キー値の大きい側）から正順マージ処理を行い、第2のプロセッサは同じ部分データ列の他方の端（キー値の小さい側）から逆順マージ処理を同時に行い、前記第1のプロセッサは前記領域の部分の一方の端から順にマージ処理の結果を書き込み、前記第2のプロセッサは前記領域の部分の他方の端から順にマージ処理の結果を書き込む。

【 0 0 2 0 】

前記分割条件は、予めソートされた第1データ列Aが2つの部分データ列A1と部分データ列A2あるいはそれ以上に分割され、予めソートされた第2データ列Bが2つの部分データ列B1と部分データ列B2あるいはそれ以上に分割される場合において、少なくとも（部分データ列A1又は部分データ列B1の末尾のデータのキー値のうちの小さい方） \geq （部分データ列A2又は部分データ列B2の頭部のデータのキー値のうちの大きい方）が成立しなければならない。

【 0 0 2 1 】

この発明に係る並列マージソート処理方法は、複数のプロセッサを含む並列プロセッサを用いて並列マージソート処理を行うための方法であって、

処理対象データを分割して複数の部分データ列を得る第1ステップと、

前記複数の部分データ列にそれぞれプロセッサを割り当てる第2ステップと、

割り当てられたプロセッサにより前記複数の部分データ列についてそれぞれ独立にソート処理を行い降順に整列された部分データ列を出力する第3ステップと

、
部分データ列を所定の分割条件のもとでそれぞれ要求された数に分割する第4ステップと、

分割された部分データ列のうちの少なくとも一部を組にする第5ステップと、

得られた組に対してプロセッサを割り当てる第6ステップと、

割り当てられたプロセッサによりマージ処理を行行い降順に整列された部分データ列を出力する第7ステップと、

マージ処理されたデータ列を前記部分データ列として前記第4ステップ乃至前記第7ステップを繰り返すステップと、を備えるものである。

【0022】

好ましくは、

前記第6ステップにおいて、前記部分データ列の組に対して2つのプロセッサが割り当てられ、

前記第7ステップにおいて、第1のプロセッサは前記部分データ列の一方の端（キー値の大きい側）から正順マージ処理を行い、第2のプロセッサは同じ部分データ列の他方の端（キー値の小さい側）から逆順マージ処理を同時に行う。

【0023】

好ましくは、前記第4ステップは、

それぞれ降順に整列された n 個のデータ列を含む2つの領域 D_1 と D_2 が与えられた場合において、領域 D_1 と D_2 をそれぞれ k 組に分割し、得られた領域 D_1 の部分データ列と領域 D_2 の部分データ列とを順次組み合わせたとき、この組に含まれる最小のキー値をもつデータが領域 D_1 と D_2 の先頭から数えて $2(n/k)$ 番目、 $4(n/k)$ 番目、 $6(n/k)$ 番目、 \dots 、 $2(k-1)(n/k)$ 番目のいずれかとなるように分割するために、

指標値（領域の先頭のデータを0とし、順次1、2、 \dots となる）が (n/k) 、 $2(n/k)$ 、 $3(n/k)$ 、 \dots 、 $(k-1)(n/k)$ であるデータを仮の境界として選ぶステップと、

前記領域 D_1 と D_2 の作業用指標変数 i と j に前記仮の境界の指標値を設定するステップと、

前記領域 D_1 の指標 i が示すデータのキー値と前記領域 D_2 の指標 j が示すデータのキー値を比較する比較ステップと、

最初の比較で前記領域 D_1 の指標 i のキー値と前記領域 D_2 の指標 j のキー値が等しくない場合、大きい方の指標値に1を加え、小さい方の指標値から1を引いた後

、前記比較ステップに分岐するステップと、

2回目以降の比較で前記領域 D_1 の指標 i のキー値と前記領域 D_2 の指標 j のキー値との間の大小関係が前回の大小関係と逆転しない場合、大きい方の指標値に1を加え、小さい方の指標値から1を引いた後、前記比較ステップに分岐するステップと、

前記領域 D_1 の指標 i のキー値と前記領域 D_2 の指標 j のキー値が等しいとき、指標 i のキー値と指標 j のキー値をそれぞれ分割の境界とするステップと、

前記領域 D_1 の指標 i のキー値と前記領域 D_2 の指標 j のキー値との間の大小関係が前回の大小関係と逆転したとき、前回の比較操作における前記領域 D_1 のキー値と前記領域 D_2 のキー値の内の大きい方の数値と、今回の比較操作における前記領域 D_1 のキー値と前記領域 D_2 のキー値の内の大きい方の数値を比較して、小さい方の数値を分割の境界とするとともに、当該数値の比較相手を他方の領域の境界とするステップと、を備える。これは正順併合のために用いる。

【 0 0 2 4 】

好ましくは、前記第4ステップは、

それぞれ降順に整列された n 個のデータ列を含む2つの領域 D_1 と D_2 が与えられた場合において、領域 D_1 と D_2 をそれぞれ k 組に分割し、得られた領域 D_1 の部分データ列と領域 D_2 の部分データ列とを順次組み合わせるとき、この組に含まれる最小のキー値をもつデータが領域 D_1 と D_2 の先頭から数えて $2(n/k)$ 番目、 $4(n/k)$ 番目、 $6(n/k)$ 番目、 \dots 、 $2(k-1)(n/k)$ 番目のいずれかとなるように分割するために、

指標値（領域の先頭のデータを0とし、順次1、2、 \dots となる）が $(n/k)-1$ 、 $2(n/k)-1$ 、 $3(n/k)-1$ 、 \dots 、 $(k-1)(n/k)-1$ であるデータを仮の境界として選ぶステップと、

前記領域 D_1 と D_2 の作業用指標変数 i と j に前記仮の境界の指標値を設定するステップと、

前記領域 D_1 の指標 i が示すデータのキー値と前記領域 D_2 の指標 j が示すデータのキー値を比較する比較ステップと、

最初の比較で前記領域 D_1 の指標 i のキー値と前記領域 D_2 の指標 j のキー値が等

しくない場合、大きい方の指標値に1を加え、小さい方の指標値から1を引いた後、前記比較ステップに分岐するステップと、

2回目以降の比較で前記領域 D_1 の指標 i のキー値と前記領域 D_2 の指標 j のキー値との間の大小関係が前回の大小関係と逆転しない場合、大きい方の指標値に1を加え、小さい方の指標値から1を引いた後、前記比較ステップに分岐するステップと、

前記領域 D_1 の指標 i のキー値と前記領域 D_2 の指標 j のキー値が等しいとき、指標 i のキー値と指標 j のキー値をそれぞれ分割の境界とするステップと、

前記領域 D_1 の指標 i のキー値と前記領域 D_2 の指標 j のキー値との間の大小関係が前回の大小関係と逆転したとき、前回の比較操作における前記領域 D_1 のキー値と前記領域 D_2 のキー値の内の小さい方の数値と、今回の比較操作における前記領域 D_1 のキー値と前記領域 D_2 のキー値の内の小さい方の数値を比較して、大きい方の数値を分割の境界とするとともに、当該数値の当初の比較相手を他方の領域の境界とするステップと、を備える。これは逆順併合に用いられる。

【0025】

この発明は、2組のソート済のデータを任意の並列度のマージソートで整列させる並列マージソート処理方法であって、

大きさの順に整列された複数のキー値をそれぞれ含む第1のデータ列及び第2のデータ列を用意するステップと、

要求される並列度に応じて前記第1のデータ列及び前記第2のデータ列をそれぞれ分割するステップと、

分割されたデータ列間で並列マージを行うステップとを備えるものである。

【0026】

この発明は、複数のプロセッサを含む並列プロセッサを用いて並列マージソート処理を行うためのプログラムであって、

処理対象データを分割して複数の部分データ列を得る第1ステップと、

前記複数の部分データ列にそれぞれプロセッサを割り当てる第2ステップと、

割り当てられたプロセッサにより前記複数の部分データ列についてそれぞれ独立にソート処理を行い降順に整列された部分データ列を出力する第3ステップと

部分データ列を所定の分割条件のもとでそれぞれ要求された数に分割する第 4 ステップと、

分割された部分データ列のうちの少なくとも一部を組にする第 5 ステップと、
得られた組に対してプロセッサを割り当てる第 6 ステップと、

割り当てられたプロセッサによりマージ処理を行い降順に整列された部分データ列を出力する第 7 ステップと、

マージ処理されたデータ列を前記部分データ列として前記第 4 ステップ乃至前記第 7 ステップを繰り返すステップと、を実行させるためのプログラムである。

【 0 0 2 7 】

この発明に係るプログラムは、例えば、記録媒体に記録される。

媒体には、例えば、EPROM デバイス、フラッシュメモリデバイス、フレキシブルディスク、ハードディスク、磁気テープ、光磁気ディスク、CD (CD-ROM、Video-CD を含む)、DVD (DVD-Video、DVD-ROM、DVD-RAM を含む)、ROM カートリッジ、バッテリーバックアップ付きの RAM メモリカートリッジ、フラッシュメモリカートリッジ、不揮発性 RAM カートリッジ等を含む。

【 0 0 2 8 】

また、電話回線等の有線通信媒体、マイクロ波回線等の無線通信媒体等の通信媒体を含む。インターネットもここでいう通信媒体に含まれる。

【 0 0 2 9 】

媒体とは、何等かの物理的手段により情報（主にデジタルデータ、プログラム）が記録されているものであって、コンピュータ、専用プロセッサ等の処理装置に所定の機能を行わせることができるものである。要するに、何等かの手段でもってコンピュータにプログラムをダウンロードし、所定の機能を実行させるものであればよい。

【 0 0 3 0 】

【発明の実施の形態】

発明の実施の形態 1.

発明の実施の形態に係る装置／方法について図面を参照して説明する。

図 1 は本装置の概略構成を示すものである。1 は複数のプロセッサを含み、複数のタスクを同時並行的に実行可能な並列プロセッサ、2 はマージソート処理の過程において中間のマージ結果（データ列）を受け、これを所定の部分で所定の数に分割して並列プロセッサ 1 に返す領域分割部である。実際には、領域分割部 1 は並列プロセッサ 1 の内部に設けられてもよいし、外部に設けられてもよい。領域分割部 1 がプロセッサの数に応じた数の領域を返すことによりより多くのプロセッサに処理を実行させることができるようになるので、プロセッサの使用効率が高まり処理時間が短くなる。後述の図 3 の例では、マージの入力となるデータ列 D 2 1 と D 2 2 の組をそれぞれ 2 分割することにより 2 組の入力データ列を得て、それぞれを正順マージと逆順マージを行うことによりプロセッサ P 1 ～ P 4 に同時に処理をさせることができ、データ列 D 3 1、D 3 2 をそれぞれ 4 分割することにより 4 組の入力データ列を得て、それぞれを正順マージと逆順マージを行うことによりプロセッサ P 1 ～ P 8 に同時に処理をさせることができる。詳しくは後述する。

【 0 0 3 1 】

図 2 は本方法の概略フローチャートである。

S 1 : 処理対象のデータをプロセッサの数 p に対応する p 区画 ($p = 2^q$ 、 q は整数) に分割する。

【 0 0 3 2 】

S 2 : p 個のプロセッサを分割された p 区画に割り当てる。

S 3 : それぞれ独立にソートする。

並列 2 分木マージソート法では、整列すべきデータを含む領域を $p = 2^q$ の関係をもつ p 区画に分割し、それぞれを独立に、例えばクイックソート法により整列させる。S 1 ～ S 3 は従来と同じ手順であり、 p 個のプロセッサに同時並行的にソート処理を行わせるための手順である。その後、以下のように q 段のマージソートを繰り返し、最終的に全体として整列された 1 組のデータ列を得る。

【 0 0 3 3 】

S 4 : マージを開始する。マージの段数は S 1 の分割による区画数に対応する。

- (1) $a = a + 1$
- (2) 2^{q-a+1} 組の部分データ列を2つずつ組み合わせた区画対を作る。
- (3) 区画対を 2^{a-1} に分割する。

【 0 0 3 4 】

正順併合と逆順併合を同時に行うので2つのデータ列のマージに2つのプロセッサを使用するためである。正順併合とは、キー値の大きい側(始端)から併合操作を始めて結果を出力域の始端から降順に出力する手法であり、降順併合とは、キー値の小さい側(終端)から併合操作を始めて出力域の終端から昇順に出力する手法である。併合対象のデータ列を格納している2つの領域を任意組の区画対、例えばk組(kは2以上の整数)の区画対に分割できたとする。分割した各区画対に含まれるデータ群に正順併合と逆順併合操作を同時並行的に実行すれば、並列度は一挙に2kにあがる。

【 0 0 3 5 】

S 5 : (2^{q-a}) (2^{a-1}) 組の区画対のそれぞれに正順併合と逆順併合用のプロセッサを割り当てる。区画対の作り方については後述する。

【 0 0 3 6 】

S 6 : 各組についてそれぞれ2つのプロセッサを用いて正順併合と逆順併合を同時に行い $2^{(q-a)}$ 組の整列された部分データ列を得る。

【 0 0 3 7 】

S 7 : 次に、 $a \neq q$ なら a に1を加え、S 4 ~ S 7 の処理を繰り返す。

順次 $a = 3, \dots, q$ として S 4 ~ S 7 の処理を繰り返す。

【 0 0 3 8 】

S 8 : 以上の手順により、最終的に全体として整列された1組のデータ列を得ることができる。

【 0 0 3 9 】

図2の方法は、2段目においてマージ対象データを含む領域を2分割し、3段目において4分割し、4段目において8分割し、 \dots 、 q 段目において領域を 2^{q-1} 分割し、それぞれの領域に正順併合と逆順併合を適用するものである。図2の領域分割法を適用したマージソート法によれば、常に p 並列の併合操作を

行うことができる。

【 0 0 4 0 】

図 3 を参照してプロセッサの数が 8 の場合における並列＝分木マージソート法の処理について説明する。図中、丸はプロセッサを示し、四角はデータ列あるいはデータ列 (D) が記憶される領域を示す。丸の中の記号はプロセッサが行う処理の内容を示し、S は整列 (ソート) 処理、M は併合 (マージ) 処理、V は転送処理を意味する。

【 0 0 4 1 】

入力領域に置かれたデータは 8 組の未整列のデータ列に分割される。8 つのプロセッサ P 1 ～ P 8 によりこれら部分について同時並行的にソート処理を行う。これらの処理結果は領域 D 1 1 ～ D 1 8 に置かれる。

【 0 0 4 2 】

次に、第 1 段のマージソートを行う。プロセッサ P 1、P 2 によりデータ列 D 1 1 と D 1 2 が併合されて領域 D 2 1 に置かれ、プロセッサ P 3、P 4 によりデータ列 D 1 3 と D 1 4 が併合されて領域 D 2 2 に置かれ、プロセッサ P 5、P 6 によりデータ列 D 1 5 と D 1 6 が併合されて領域 D 2 3 に置かれ、プロセッサ P 7、P 8 によりデータ列 D 1 7 と D 1 8 が併合されて領域 D 2 4 に置かれる。なお、第 1 段において正順併合と逆順併合が同時に行われるので、領域分割を行わなくても全てのプロセッサが使用される (プロセッサの割り当ては便宜上のものである。以下同じ)。

【 0 0 4 3 】

次に、第 2 段のマージソートを行う。全てのプロセッサを使用するために領域分割を行う。すなわち、部分データ列 D 2 1 と D 2 2 を対にしてキー値の大小を考慮して 2 分割し、D 2 1 - 1 と D 2 2 - 1 の対と D 2 1 - 2 と D 2 2 - 2 の対を作る。データ列 D 2 3、D 2 4 についても同様である。

【 0 0 4 4 】

そして、プロセッサ P 1、P 2 によりデータ列 D 2 1 - 1 と D 2 2 - 1 が併合されて領域 D 3 1 の一方 (キー値の大きい側) に置かれ、プロセッサ P 3、P 4 によりデータ列 D 2 1 - 2 と D 2 2 - 2 が併合されて領域 D 3 1 の他方 (キー値

の小さい側)に置かれる。同様に、プロセッサP5～P8によりデータ列D23とD24が併合されて領域D32に置かれる。

【0045】

次に、第3段のマージソートを行う。第2段階と同様に領域分割が行われる。すなわち、データ列D31とD32を対にし、キー値の大小を考慮して4分割し、D31-1とD32-1、D31-2とD32-2、D32-3とD32-3、D31-4とD32-4の対を作る。プロセッサP1、P2によりデータ列D31-1とD32-1が併合されて領域D4内の所定の位置に置かれ、プロセッサP3、P4によりデータ列D31-2とD32-2が併合されて領域D4内の所定の位置に置かれ、プロセッサP5、P6によりデータ列D31-3とD32-3が併合されて領域D4内の所定の位置に置かれ、そして、プロセッサP7、P8によりデータ列D31-4とD32-4が併合されて領域D4内の所定の位置に置かれる。これでマージソートが完了する。得られた結果D4は8つのプロセッサにより出力領域に転送される。

【0046】

図3の処理の各段階において8つのプロセッサの全てが使用される。

図4～図6を参照して図3の処理を模式的に説明する。

d10～d13、d20～d23の8ブロック（各ブロックのデータ数＝380個）の合計3040組のデータをソートすることを考える。

(1) d10～d23のそれぞれを8台のプロセッサを使いソートする。ソートされた状態は図4の中段のようにキー値が降順に並べられたものになる。図中、三角形は各ブロックにおけるキー値の大きさの分布を示す。

(2) 別に一組の作業領域e10～e13、d20～d23を設けて図4の中段のように並列マージを実行する。その結果、下段のような結果が得られる。

(3) 次に2段目のマージソートを行う。(e10、e11)と(e20、e21)を組み合わせ、キー値の大小を考慮して2分割して2組のマージ入力対を作り4並列のマージを実行する(図5上段)。同時に、(e12、e13)と(e22、e23)を組み合わせ、キー値の大小を考慮して2分割して2組のマージ入力対を作り4並列のマージを実行する(図5下段)。結局、4並列のマージが2系統並列に実行

される。

(4) 次に3段目のマージソートを行う。d10～d13とd20～d23を組み合わせ、e10～e23へ出力する。各領域は4分割され、それぞれ正順・逆順の処理がされるので、8つの処理が並列に実行される。

【0047】

図3の各段階におけるマージソート処理についてさらに詳しく説明する。図7はその説明図である。入力領域 (D_1, n) と (D_2, n) が与えられたとする。ここで n はデータ数を意味する。図7の各領域において左側ほどキー値が大きく、右側ほどキー値が小さくなるようにソートされている。

【0048】

領域分割部2により領域 (D_1, n) のデータ列が (D_{11}, n_{11}) と領域 (D_2, n) の整列された部分データ列とを対にしキー値の大小を考慮して2分割し、 (D_{11}, n_{11}) 、 (D_{21}, n_{21}) の対と (D_{12}, n_{12}) と (D_{22}, n_{22}) の対とに分けられたとする。ここで、 $(D_{11}$ の末尾のデータのキー値) $\geq (D_{12}$ の頭部のデータのキー値)、 $n_{11} + n_{12} = n$ 、 $(D_{21}$ の末尾のデータのキー値) $\geq (D_{22}$ の頭部のデータのキー値)、 $n_{21} + n_{22} = n$ である。

【0049】

領域 (D_{11}, n_{11}) と (D_{21}, n_{21}) のマージ入力の対に対して、プロセッサP1で正順併合を行うとともに、プロセッサP2で逆順併合を行う。図中の点線は併合の順序を示す。その結果を出力領域 $(D_3, 2n)$ の部分である (D_{31}, n_{31}) に順次出力する。同様に、領域 (D_{12}, n_{12}) と (D_{22}, n_{22}) に対して、プロセッサP3で正順併合を行うとともに、プロセッサP4で逆順併合を行う。その結果を出力領域 $(D_3, 2n)$ の部分である (D_{32}, n_{32}) に順次出力する。ここで、 $(D_{31}$ の末尾のデータのキー値) $\geq (D_{32}$ の頭部のデータのキー値)、 $n_{31} + n_{32} = 2n$ である。

【0050】

以上のように入力領域対を分割するとともに分割された領域それぞれにプロセ

ッサを割り当てることにより、複数のプロセッサによる同時並行処理が可能になる。なお、以上の説明はあくまで一例であり、本発明はこれに限定されない。

【 0 0 5 1 】

次に領域の分割手順について説明する。図 7 の例からも明らかなようにマージソートが正しく行われるためには、 $(D_{31} \text{の末尾のデータのキー値}) \geq (D_{32} \text{の頭部のデータのキー値})$ でなければならない。 D_{31} は D_{11} 、 D_{21} のマージソート結果であり、 D_{32} は D_{12} 、 D_{22} のマージソート結果である。したがって、 $(D_{11} \text{または} D_{21} \text{に含まれるデータのキー値の最小値}) \geq (D_{12} \text{または} D_{22} \text{に含まれるデータのキー値の最大値})$ ということである。このことから本発明の領域分割の条件は次のようなものである。

【 0 0 5 2 】

「予めソートされた第 1 データ列 A と B を対にしキー値の大小を考慮して (A_1, B_1) 、 (A_2, B_2) に分割される場合において、 $(\text{部分データ列} A_1 \text{ 又は部分データ列} B_1 \text{ に含まれるデータのキー値の最小値}) \geq (\text{部分データ列} A_1 \text{ 又は部分データ列} B_1 \text{ に含まれるデータのキー値の最大値})$ 、 $(\text{部分データ列} A_2 \text{ と} B_2 \text{ に含まれるデータのキーの最小値}) \geq (\text{部分データ列} A_2 \text{ と} B_2 \text{ に含まれるデータのキーの最小値})$ を満足すること」

【 0 0 5 3 】

以上の条件を満たす領域分割の手法を説明する。以下の正順併合用の手法又は逆順併合用の手法のいずれかを用いることにより、上記条件を満たす領域分割を実現することができる。

【 0 0 5 4 】

領域分割手法の具体的な手順の説明の前に前提となる事項について説明する。

主記憶の二つの領域 (D_1 域と D_2 域) に降順に整列された各 n 組のデータを仮定し、両領域を (n/k) 組のデータを含む k 組ずつの区画に分割する。

個々の区画は 0 から $k-1$ の区画番号で指定する。各データの位置は、領域頭部を 0 としデータごとに 1 ずつ増加する指標値で指定する。この位置指定法は D_1 域、 D_2 域全体のデータ位置のほか、分割した各領域内のデータ位置の指定にも用いる。

また、領域の頭部あるいは尾部という表現で始端あるいは終端のデータ位置を指定する。

分割された D_1 域と D_2 域の領域中、同一区画番号の領域同志を仮の併合対象領域対とする。

【 0 0 5 5 】

すなわち、区画番号 i ($i=0\sim k-1$)の領域を対象とする正順併合の始点は、 D_1 域と D_2 域の始端から $(n/k)(i)$ 組のデータを除いた残りの領域の始端、すなわち指標値が $(n/k)(i)$ の点とする。また、区画番号 i ($i=0\sim k-1$)の領域を対象とする逆順併合の始点は、 D_1 域と D_2 域の始端からそれぞれ $(n/k)(i+1)$ 組のデータが存在する領域の末尾、換言すれば指標値が $(n/k)(i+1)-1$ の点とする。

D_1 域と D_2 域内のキー値の分布は互いに独立である。このため、それぞれの領域内で整列済であっても、仮の開始点から併合操作をしたのでは、全体としてキー値の分布に矛盾のない整列結果を得ることはできない。そこで以下の手順により前記条件を満たす開始点（領域分割点）を求める。

【 0 0 5 6 】

<正順併合の場合>

それぞれ降順に整列された n 組のデータ列を収容した2つの領域、 D_1 域と D_2 域とがある。段落 0 0 4 8 の条件を満たし、かつ、両領域のキー値の大小分布を考慮して領域先頭からのデータ数が $2(n/k)$ 、 $4(n/k)$ 、 $6(n/k)$ 、 \dots 、 $2(k-1)(n/k)$ となるように分割する。

図 8 を参照する。

S 1 0 : 境界初期値（あるいは仮の境界値）として (n/k) 、 $2(n/k)$ 、 $3(n/k)$ 、 \dots 、 $(k-1)(n/k)$ の指標値をもつ点を選ぶ。

S 1 1 : D_1 と D_2 域の作業用指標 i と j とに、S 1 0 で定めた指標値を設定する。

S 1 2 : 下記の手順を大小関係が反転する迄繰り返す。ただし最初の比較を含め、キー値が等しい場合も反転とみなす。

S 1 2 a : $D_1(i)$ と $D_2(j)$ のキー値を比較する。

S 1 2 b : 大小関係を判別する。

(1)両者が等しいなら S 1 3 a に分岐する。

(2) 大小関係が逆転したら S 1 3 b に分岐する。

(3) 最初の比較で両者が等しくない場合、または2回目以降の比較で大小関係が反転しない場合、キー値の大きい方の指標値に1を加え、小さい方の指標値から1を引いた後、S 1 2 に分岐する (S 1 2 c)。

S 1 3 a : i と j を境界点の指標とし処理を終わる。

S 1 3 b : 大小関係が変化する直前と直後の比較操作の対象データ中、大きいキー値同志を比較し、小さい方を境界のデータとする。他方の領域の境界データには、上で求めたデータの当初の比較相手を選ぶ。

【 0 0 5 7 】

<逆順併合の場合>

それぞれ降順に整列された n 組のデータ列を収容した2つの領域、 D_1 域と D_2 域とがある。段落 0 0 4 8 の条件を満たし、かつ、両領域のキー値の大小分布を考慮して領域先頭からのデータ数が $2(n/k)$ 、 $4(n/k)$ 、 $6(n/k)$ 、 \dots 、 $2(k-1)(n/k)$ となるように分割する。

S 2 0 : 境界初期値 (あるいは仮の境界値) として $(n/k)-1$ 、 $2(n/k)-1$ 、 $3(n/k)-1$ 、 \dots 、 $(k-1)(n/k)-1$ の指標値をもつ点を選ぶ。

S 2 1 : D_1 と D_2 域の作業用指標変数 i と j とに、基準境界点の指標値を設定する。

S 2 2 : 下記の手順を大小関係が反転する迄繰り返す。ただし最初の比較を含め、キー値が等しい場合も反転とみなす。

S 2 2 a : $D_1(i)$ と $D_2(j)$ のキー値を比較する。

S 2 2 b : 大小関係を判別する。

(1) 両者が等しいなら S 2 3 a に分岐する。

(2) 大小関係が反転したら S 2 3 b に分岐する。

(3) 最初の比較で両者が等しくない場合、または2回目以降の比較で大小関係が反転しない場合、キー値の大きい方の指標値に1を加え、小さい方の指標値から1を引いて S 2 2 に分岐する。

S 2 3 a : i と j を境界点の指標とし処理を終わる。

S 2 3 b : 大小関係が変化する直前と直後の比較操作の対象データ中、大きいキー値同志を比較し、その小さい方を境界のデータとする。また、他方の領域の境

界のデータには上で求めたデータの当初の比較相手を選ぶ。

【 0 0 5 8 】

上記手順による領域分割の例について説明する。

図 1 0 は降順に整列された、それぞれ10個の数値を含む下記の 2 つのデータ列を示す。図 1 0 (a) は正順併合の始点の決定手順の説明図、図 1 0 (b) は逆順併合の始点の決定手順の説明図、図 1 0 (c) は領域の分割結果を示す。

D 1 : (1948,1868,1862,1148,740,620,588,256,112,100)

D 2 : (1996,1992,1966,1694,1544,1448,1182,432,254,30)

【 0 0 5 9 】

<正順併合の始点の決定>

(1) 領域頭部から5組のデータを除いた領域の頭部が仮の分割点となる。作業用指標 i , j にこの値を設定する。 $i=j=5$ となる。

(2) $D_1(i=5)=620$ と $D_2(j=5)=1466$ を比較する。1466が大きいので、 $i=i-1=4$, $j=j+1=6$ とする。

(3) $D_1(i=4)=740$ と $D_2(j=6)=1182$ を比較する。1182が大きいので、 $i=i-1=3$, $j=j+1=7$ とする。

(4) $D_1(i=3)=1148$ と $D_2(j=7)=432$ を比較する。1148が大きいので、ここで大小関係の反転を認識する。

(5) 反転前後の比較で値の大きい1148と1182を比較し、その小さい方である1148の点を一方の境界とする。他方の境界は1148の当初の比較相手である432を得る。次式を用い、 $2 \times 5 - 3 = 7$ として他方の境界指標値を求めることもできる。

他方の開始点の指標値 = 仮併合開始点の指標 $\times 2$ - 既知の開始点の指標

【 0 0 6 0 】

<逆順併合の始点の決定>

(1) 領域頭部から5組のデータがある点が仮の分割点となる。作業用指標 i , j にこの値を設定する。 $i=j=4$ とする。

(2) $D_1(i=4)=740$ と $D_2(j=4)=1544$ とを比較すると、1544が大きいので、 $i=i-1=3$, $j=j+1=5$ とする。

(3) $D_1(i=3)=1148$ と $D_2(j=5)=1446$ とを比較すると、1446が大きいので $i=i-1=2$, j

=j+1=6とする。

(4) $D_1(i=2)=1862$ と $D_2(j=6)=1182$ とを比較すると、1862が大きいので大小関係が反転したことになる。

(5) 反転前後の比較で値の小さい1148と1182を比較し、大きい方の1182を一方の境界とする。他方の境界は1182の当初の比較相手のデータ1862を得る。式11を用い、 $2 \times 4 - 6 = 2$ として他方の境界の指標値を求めることもできる。

【 0 0 6 1 】

他の例を図11(a)(b)に示す。図11(a)は、 $n=12$ 、 $k=4$ 、 $n/k=3$ の場合である。図11(b)は、 $n=10$ 、 $k=5$ 、 $n/k=2$ の場合である。この例のように、分割後の部分データ列の部分が D_1 、 D_2 の一方にのみ存在する場合（両方にまたがっていない場合）であっても、前記条件を満たすことがわかる。

本発明の実施の形態に係る装置／方法において、分割する区画の数は使用可能な処理装置の数に応じて任意に決めることができる。

【 0 0 6 2 】

本発明に係る装置／方法において、マージソートの入力となる部分データ列の対を複数の区画対に分割する時間の割合を小さくするため、データ数が大きいほど効果も大きい。

なお、複数の区画対に分割する処理も、マージソート作業を割り当てられたプロセッサ自身が自分の受け持ち領域分を実行することにより並列処理が可能となり全処理時間に占める領域対分割にかかる時間の割合を削減できる。

(1) 効果の大きい場合

D_1 域と D_2 域のキー値の分布がほぼ同じ場合は短時間に大小関係が反転する。このためマージ入力となるデータ境界の決定に多くの時間を要しない。

(2) 領域対分割に時間のかかる場合

領域対への分割に時間を要する場合として下記のケースが考えられる。

(a) 分割対の確定に時間のかかる場合

k 区画に分割する場合で対象となる領域の区画番号が $k/2$ に近い場合で、 D_1 域または D_2 の頭部や尾部、換言すれば領域の端部近辺になってようやく大小

関係が反転する場合、大小関係の反転までに時間がかかる。キーの分布がD₁域とD₂域とで偏りがある場合に発生するが、このようなケースが頻発することは少ないと考える。

(b) キー値の価域に重複がない場合。大小関係が反転しない場合

大小関係の反転を求めて比較を続け、領域の頭部または端部に至っても反転しない場合である。このときは次のように扱うことができる。すなわち、頭部に達しても反転しない場合は頭部の外側に無限大のキーを仮定して強制的に反転させ、尾部に至っても反転しない場合は尾部の外に無限小のキーを仮定して反転させる。

【 0 0 6 3 】

なお、上記例ではD₁域とD₂域に同数のデータがあるものとしていた。本発明はこれに限定されない。整列タスクや併合タスクにおいてデータ数を指定しさえすれば、D₁域とD₂域のデータの数異なる場合にも適用できる。

【 0 0 6 4 】

本発明は、主記憶を共有する並列プロセッサ方式の電子計算機に適用可能である。汎用的な技術であり、機種を問わず適用できると考えられる。本発明は、並列プロセッサ方式の電子計算機のソフトウェア（OS）に組み込まれるプログラムのひとつとして実現されると考えられる。

【 0 0 6 5 】

発明の実施の形態 2.

発明の実施の形態 1 の装置／方法は、図 2 に示すように正順併合と逆順併合を同時に用いるものであった。この発明は正順併合と逆順併合を同時に用いない場合にも適用できる。そのフローチャートを図 1 2 に示す。この処理の内容は発明の実施の形態 1 の場合と同様であるので、その説明を省略する。

【 0 0 6 6 】

本発明は、以上の実施の形態に限定されることなく、特許請求の範囲に記載された発明の範囲内で、種々の変更が可能であり、それらも本発明の範囲内に包含されるものであることは言うまでもない。

【 0 0 6 7 】

また、本明細書において、手段とは必ずしも物理的手段を意味するものではなく、各手段の機能が、ソフトウェアによって実現される場合も包含する。さらに、一つの手段の機能が、二つ以上の物理的手段により実現されても、若しくは、二つ以上の手段の機能が、一つの物理的手段により実現されてもよい。

【図面の簡単な説明】

【図 1】 発明の実施の形態に係る並列マージソート処理装置の概要を示す図である。

【図 2】 発明の実施の形態 1 に係る処理フローチャートである。

【図 3】 発明の実施の形態に係る処理全体の説明図である。

【図 4】 発明の実施の形態に係る処理の説明図である。

【図 5】 発明の実施の形態に係る処理の説明図である。

【図 6】 発明の実施の形態に係る処理の説明図である。

【図 7】 発明の実施の形態に係る処理の一部の説明図である。

【図 8】 発明の実施の形態に係る領域分割のフローチャートである（正順併合）。

【図 9】 発明の実施の形態に係る領域分割のフローチャートである（逆順併合）。

【図 1 0】 発明の実施の形態に係る領域分割の説明図である。

【図 1 1】 発明の実施の形態に係る領域分割の例を示す図である。

【図 1 2】 発明の実施の形態 2 に係る処理フローチャートである。

【図 1 3】 従来の並列マージソート処理のフローチャートである。

【図 1 4】 従来の並列マージソート処理の説明図である。

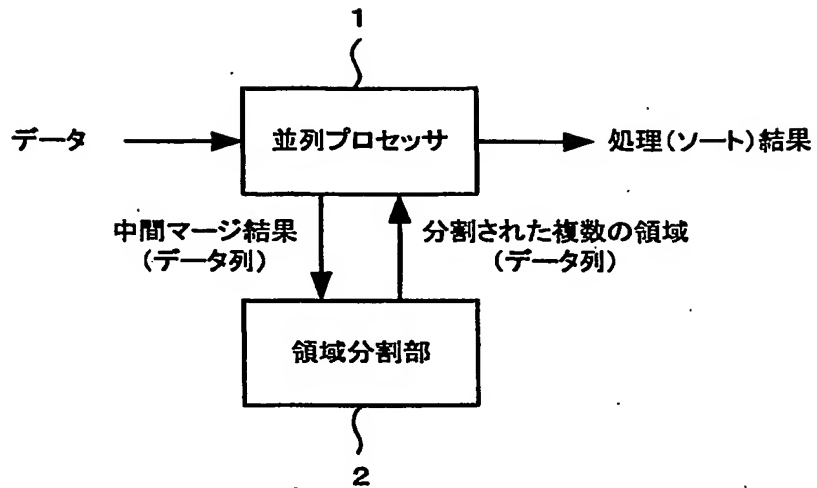
【符号の説明】

- 1 並列プロセッサ
- 2 領域分割部

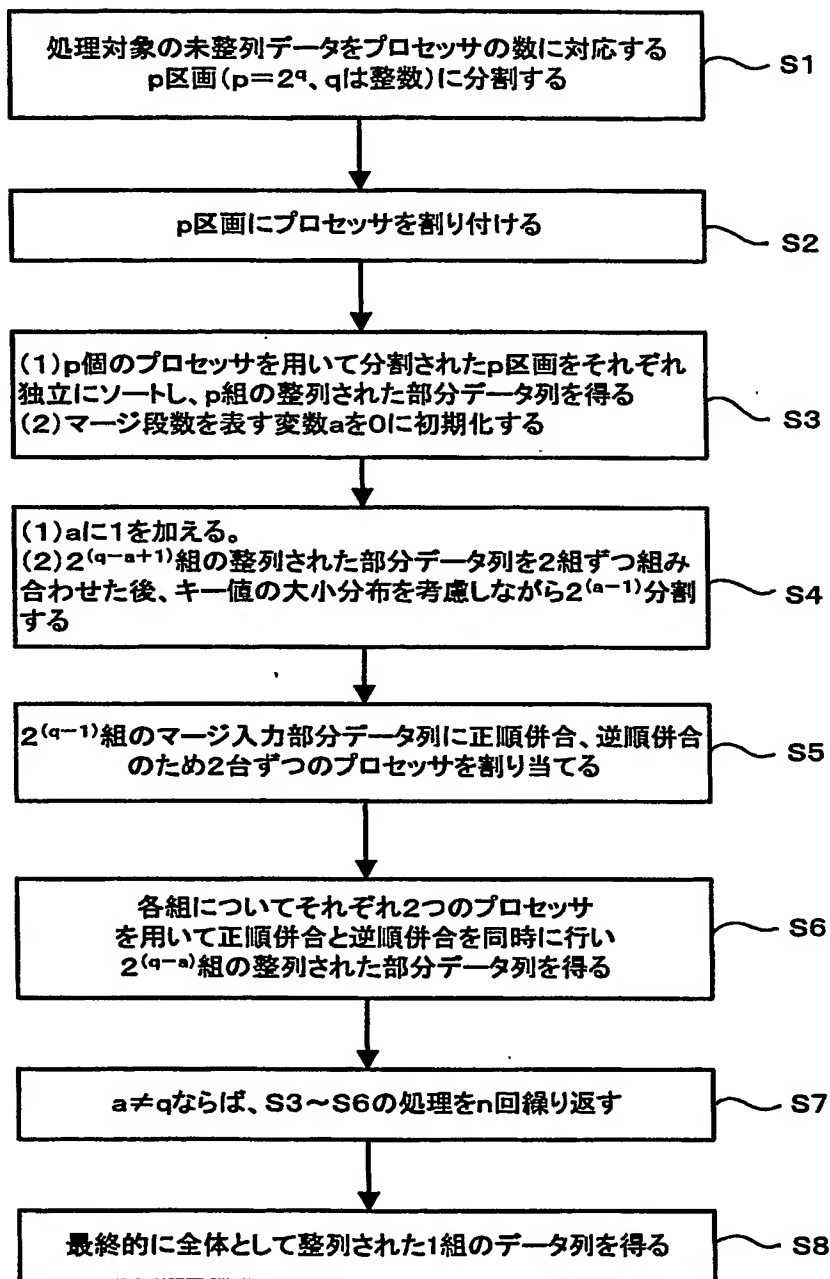
【書類名】

図面

【図 1】

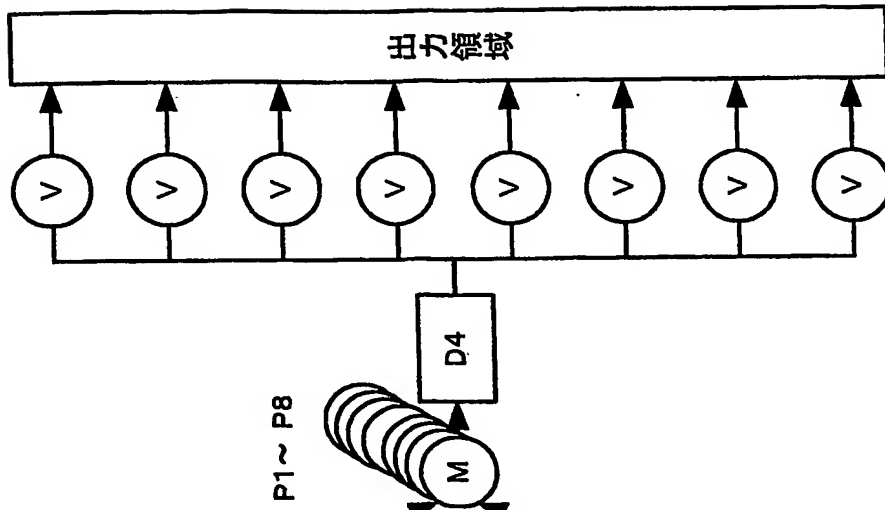


【図 2】

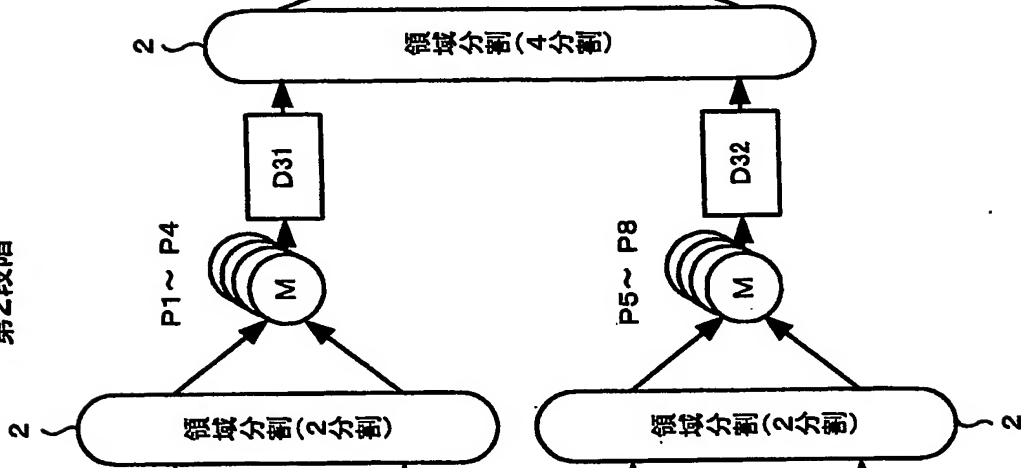


【図 3】

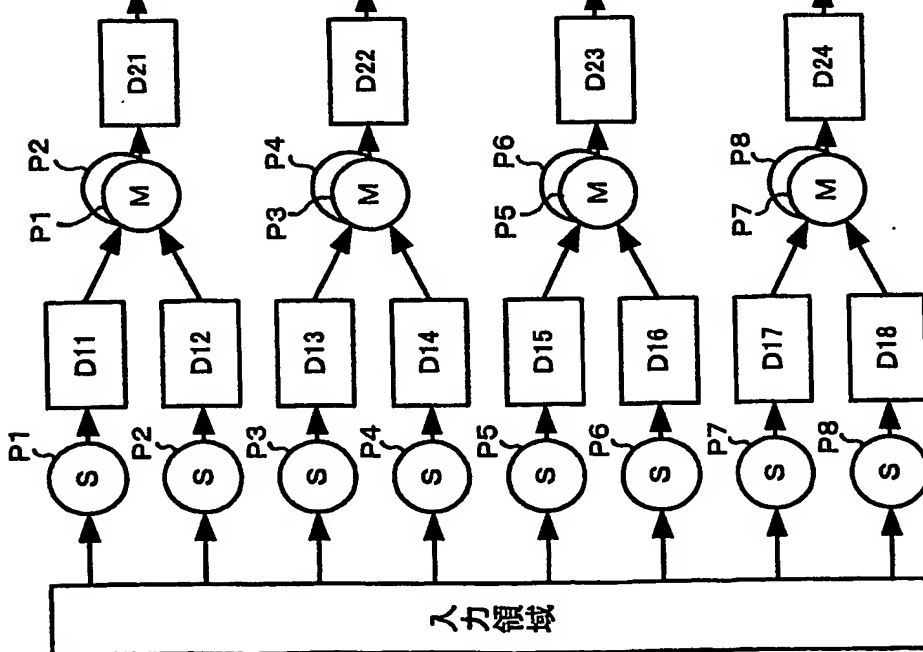
第3段階



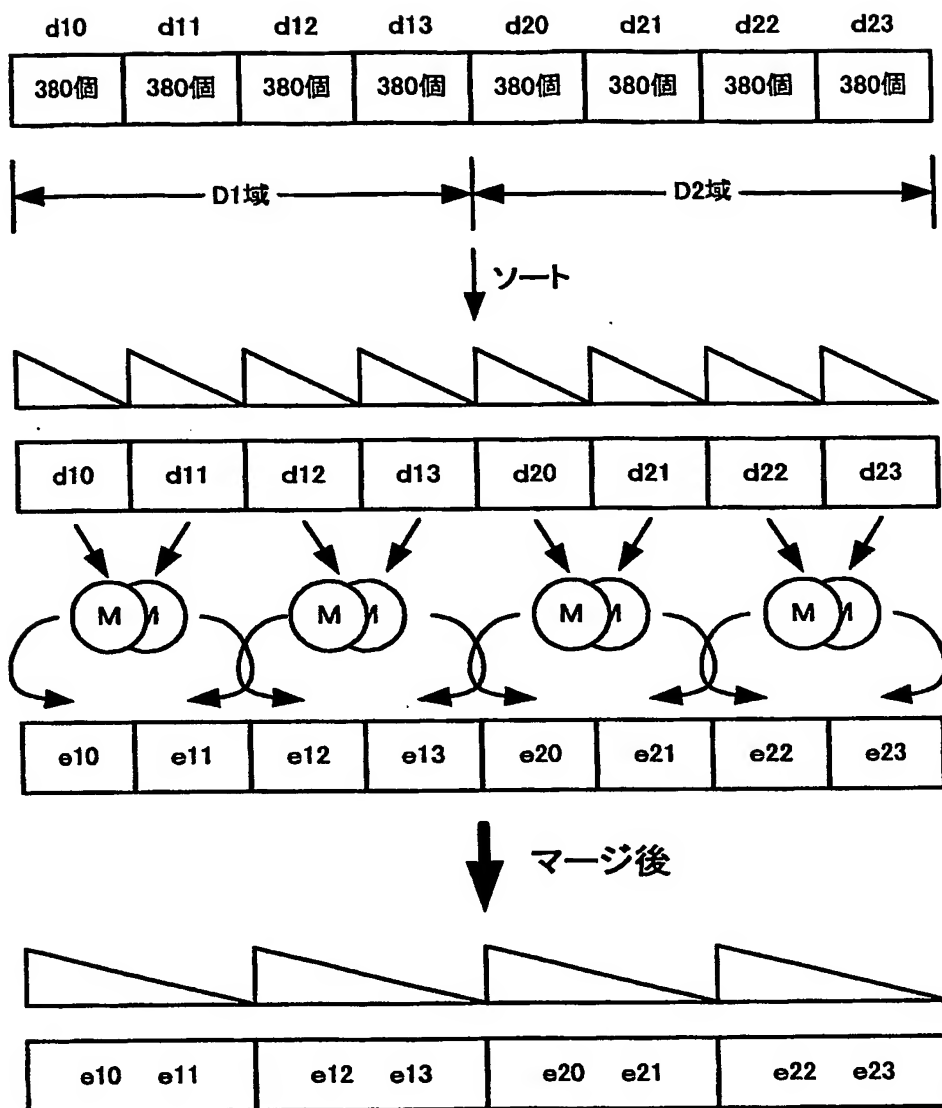
第2段階



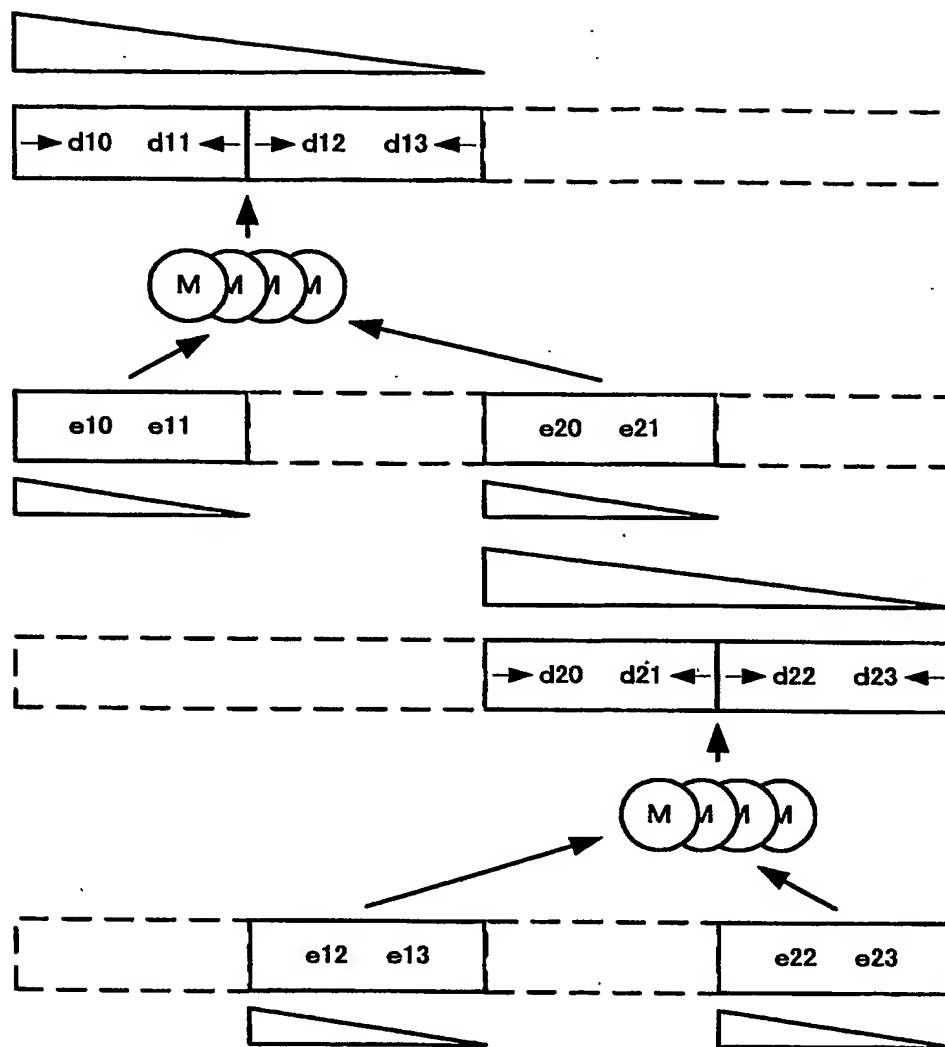
第1段階



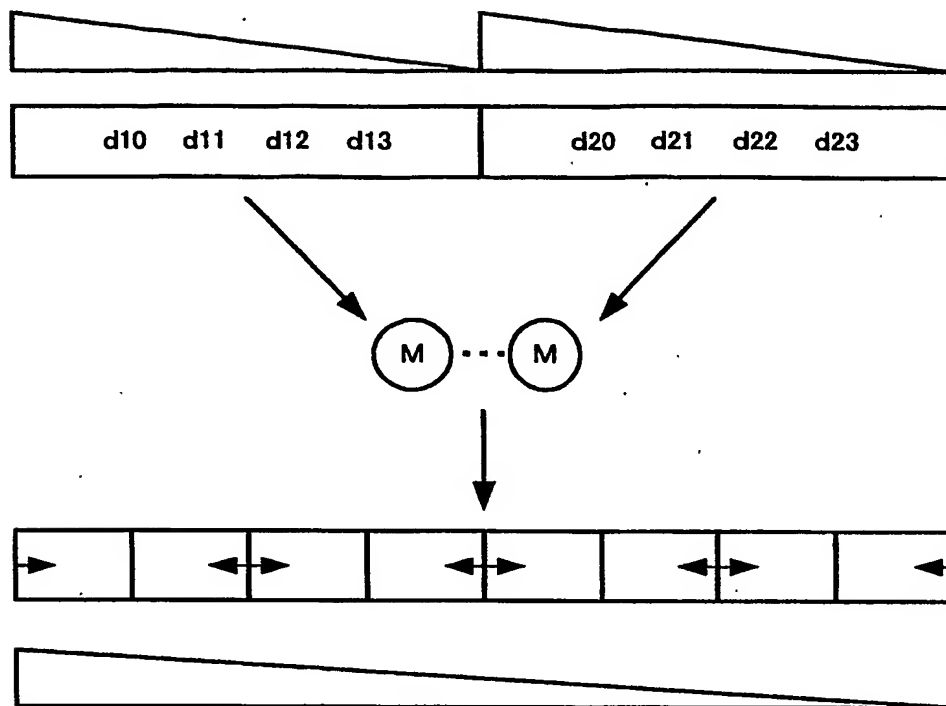
【図4】



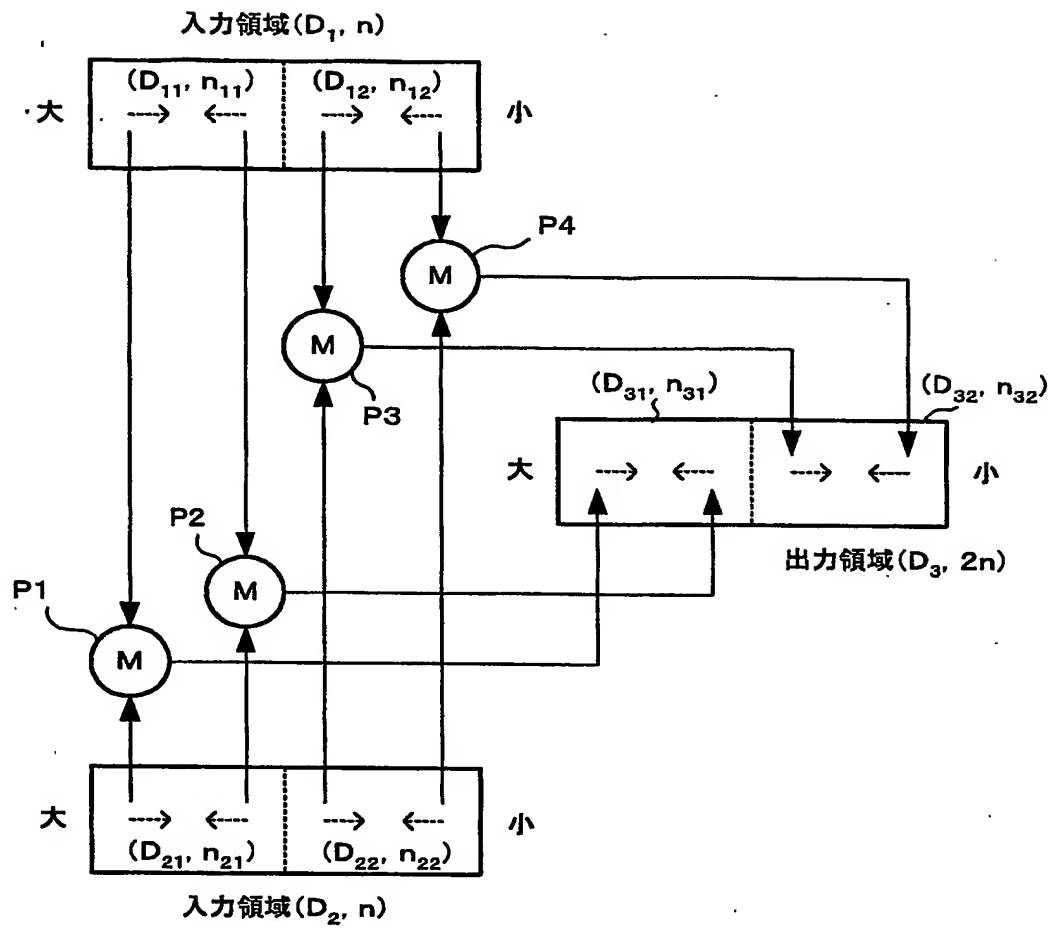
【図 5】



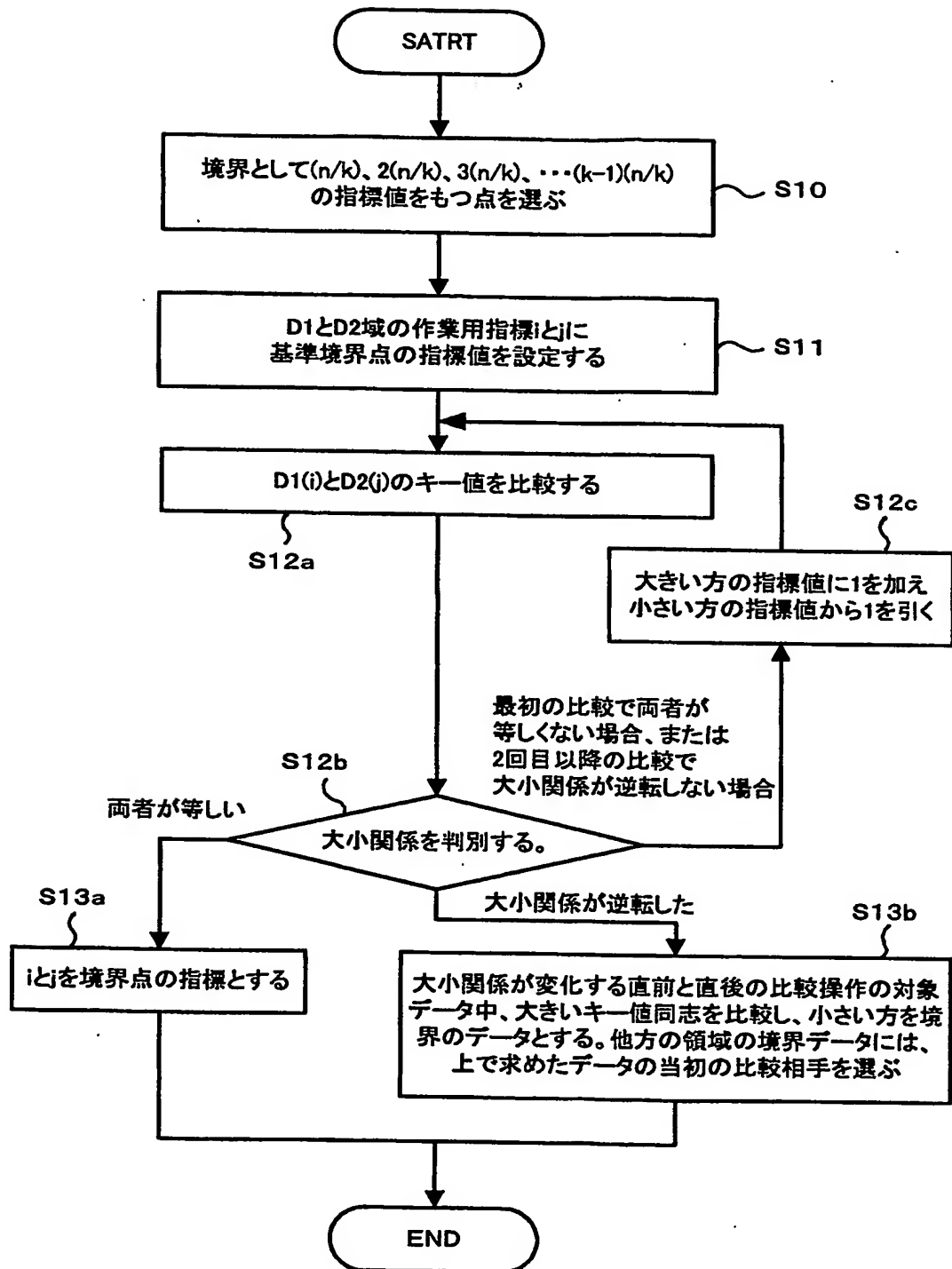
【図 6】



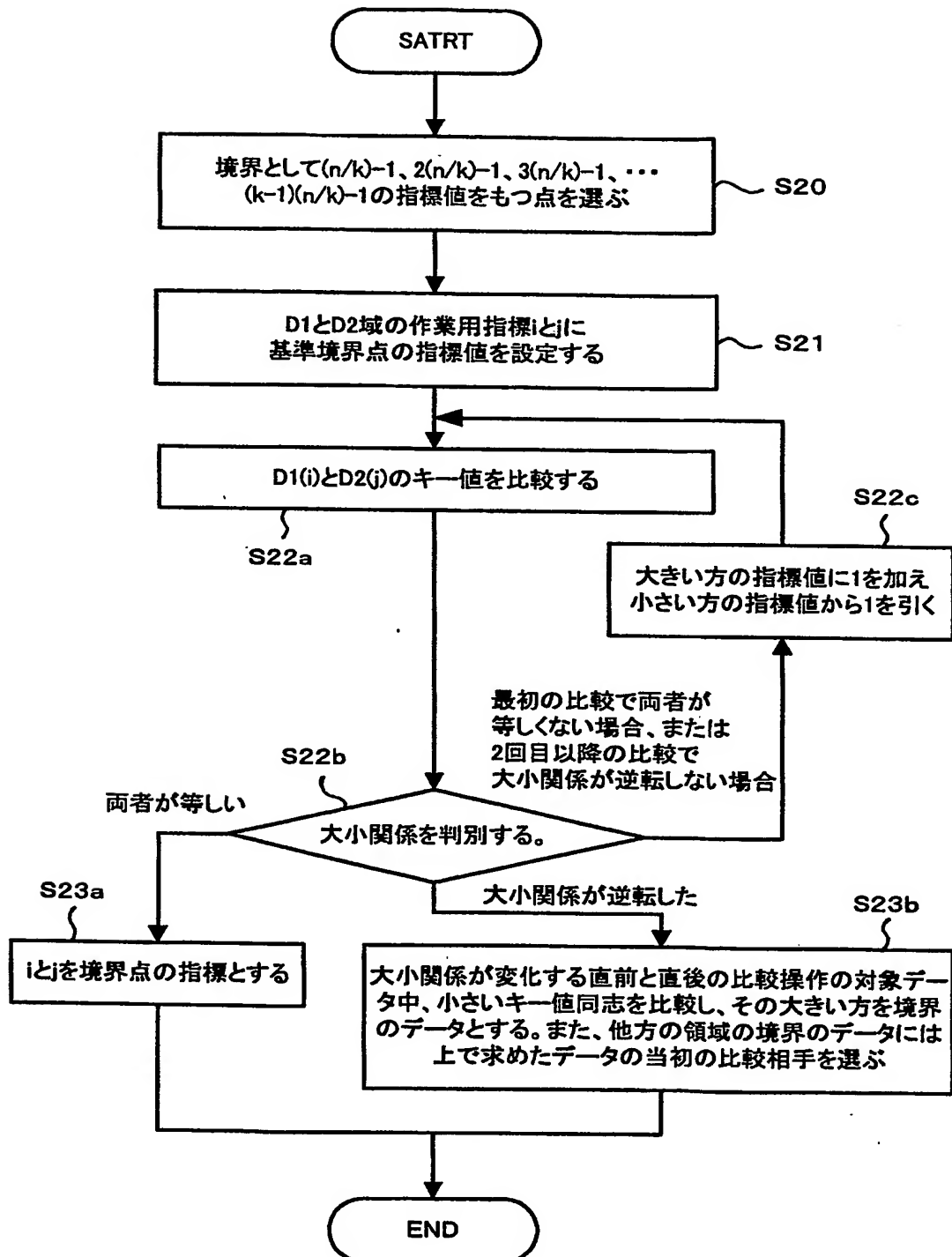
【図 7】



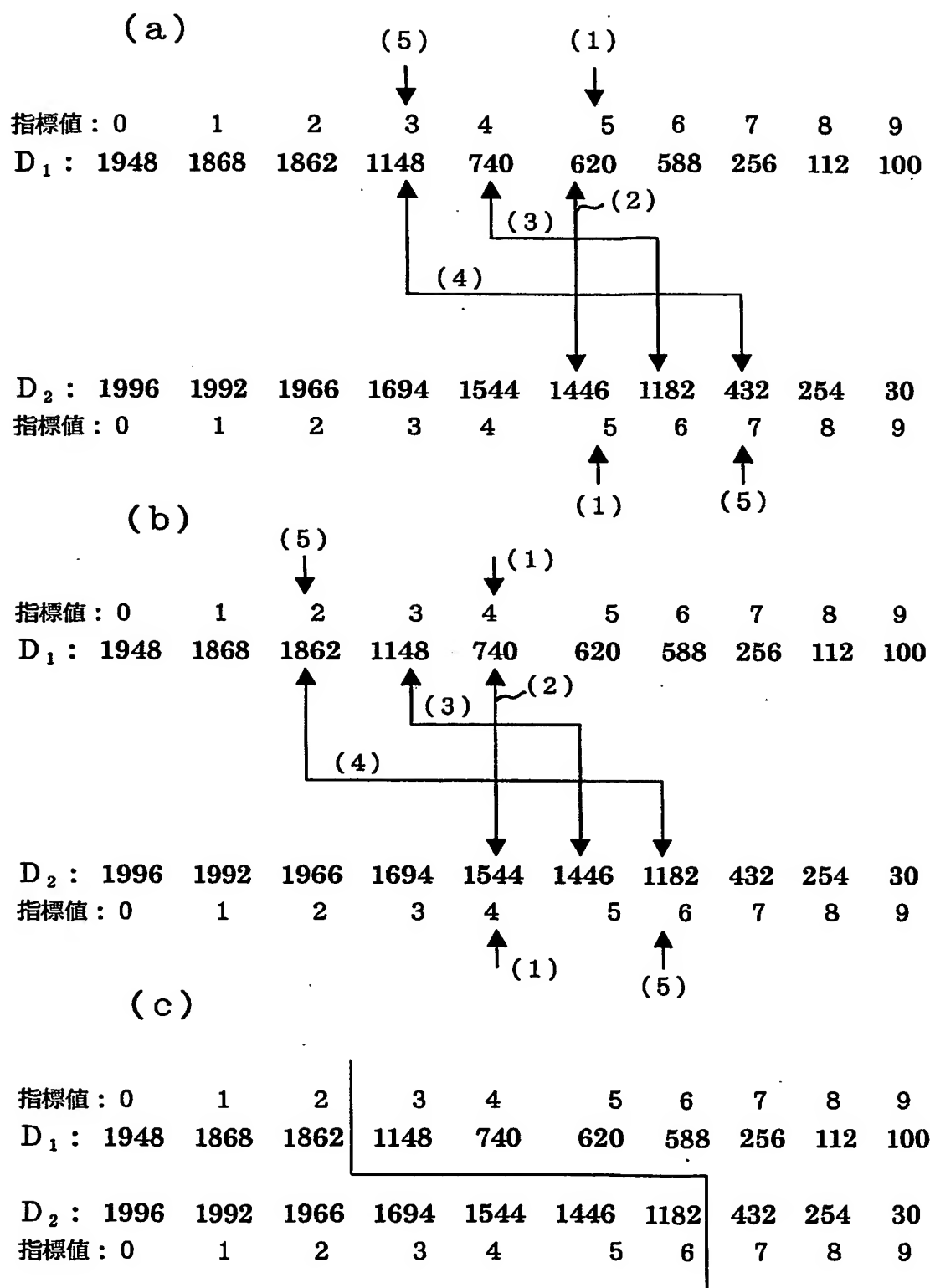
【図 8】



【図 9】



【図 1 0】



【図 1 1】

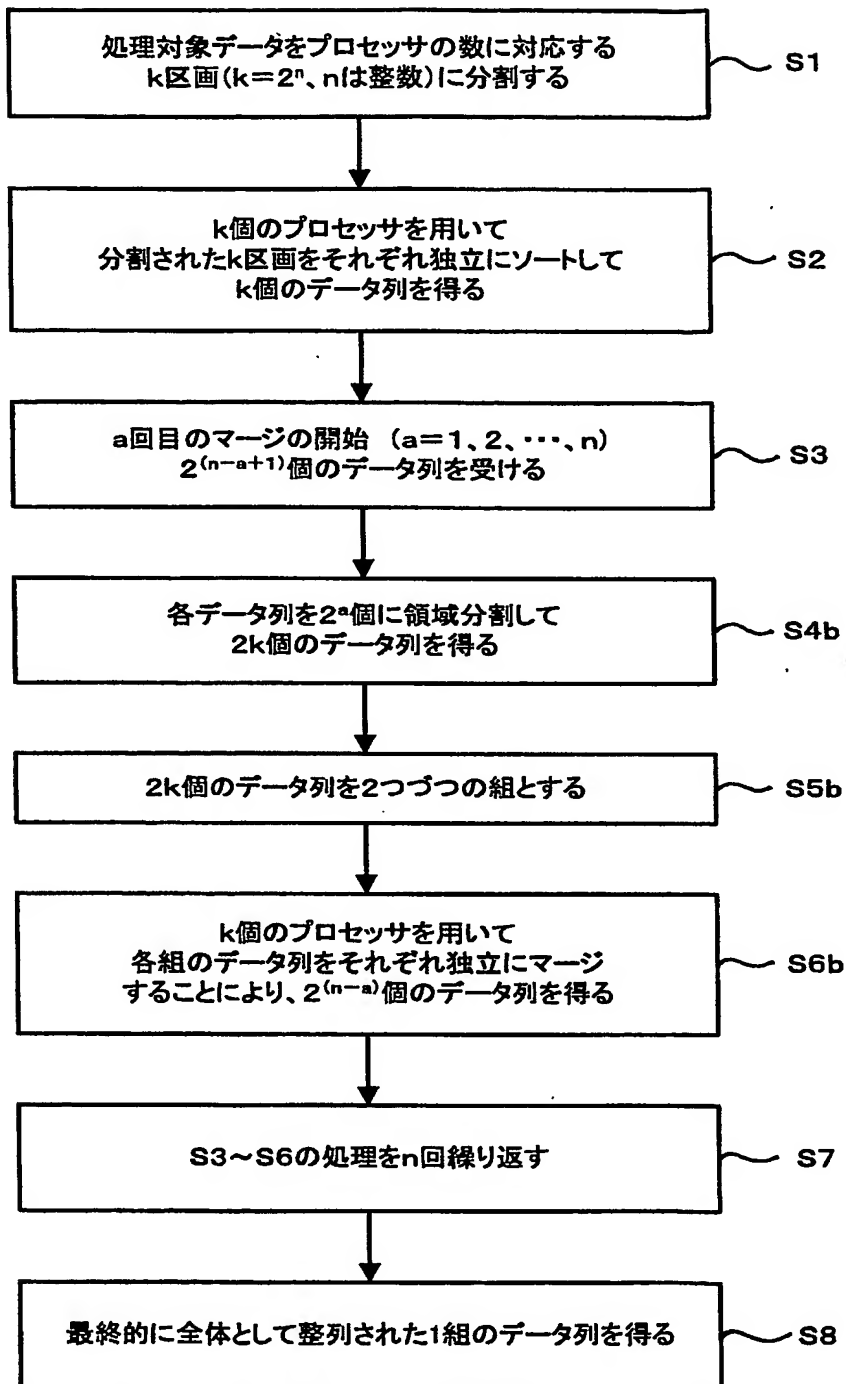
(a)

指標値 : 0	1	2	3	4	5	6	7	8	9	10	11
D ₁ : 100	99	96	90	79	76	74	72	69	65	63	61
D ₂ : 98	97	89	86	84	82	81	80	77	73	67	66
指標値 : 0	1	2	3	4	5	6	7	8	9	10	11

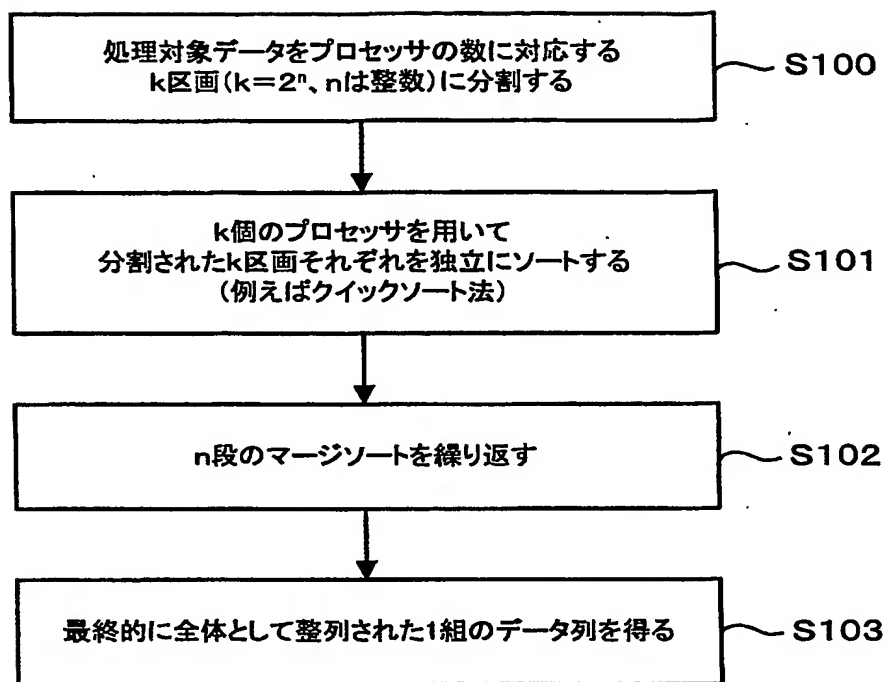
(b)

指標値 : 0	1	2	3	4	5	6	7	8	9
D ₁ : 100	98	97	93	90	89	88	87	86	60
D ₂ : 99	95	89	80	75	70	69	67	65	63
指標値 : 0	1	2	3	4	5	6	7	8	9

【図 1 2】



【図 1 3】

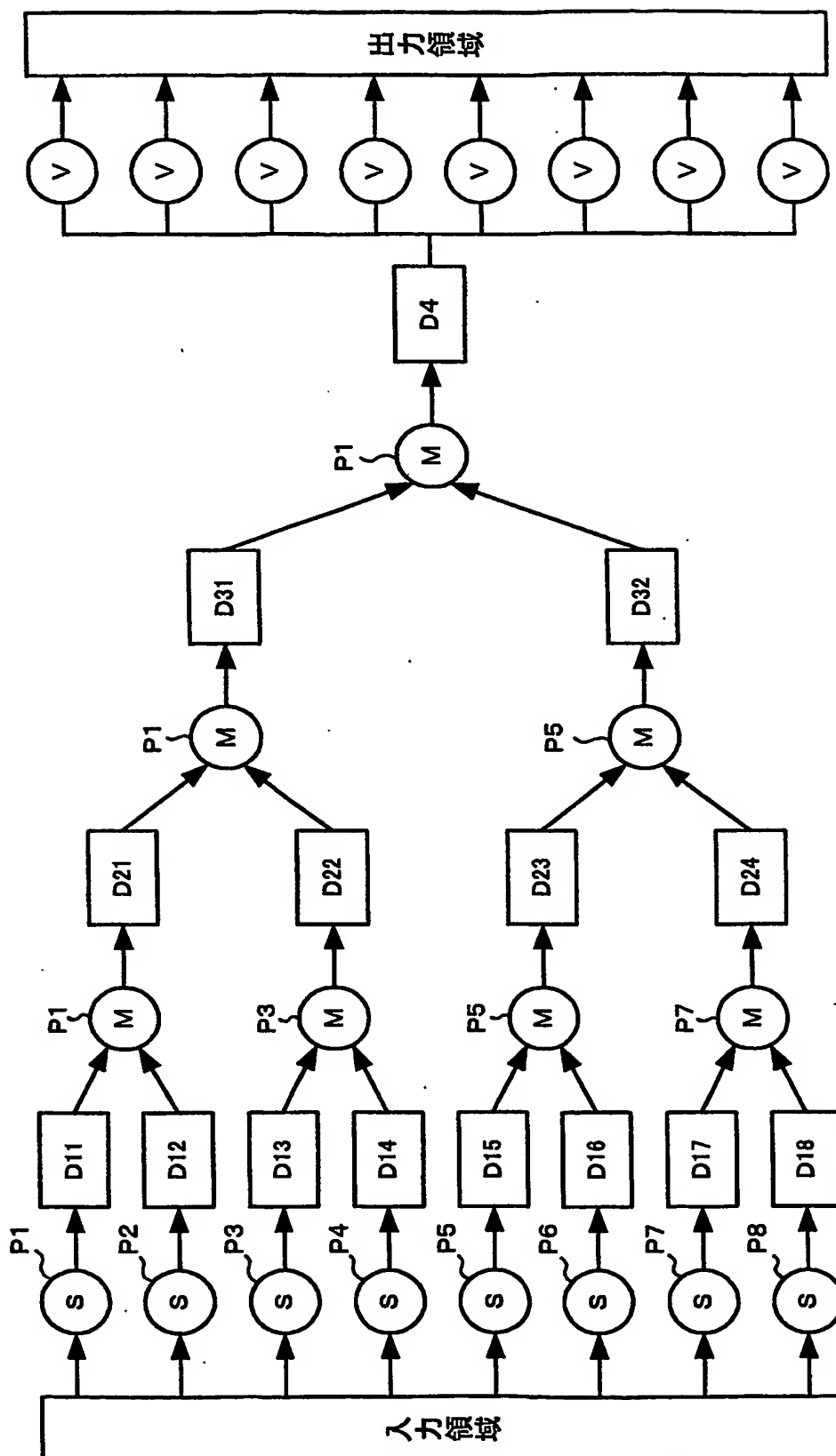


【図 1 4】

第3段階

第2段階

第1段階



【書類名】 要約書

【要約】

【課題】 複数のプロセッサを並列に接続してなる並列計算機において、各プロセッサの使用効率を高めることにより処理速度を向上させる。

【解決手段】 並列計算機を用いて行う処理のひとつであるソート処理は、入力データを分割してそれぞれについてソート処理を行うとともに、得られた複数のソート結果をマージしてひとつにまとめることにより最終的な処理結果を得る。本発明によればマージの中間過程においてマージすべきデータ（ソート結果）を分割し、これらに全てのプロセッサを割り当てることにより全てのプロセッサを効率良く使用できる。並列計算機の実行能力が向上する。

【選択図】 図 1

特 2002-125838

認定・付加情報

特許出願の番号	特願2002-125838
受付番号	50200619119
書類名	特許願
担当官	塩野 実 2151
作成日	平成14年 5月 2日

<認定情報・付加情報>

【提出日】	平成14年 4月26日
-------	-------------

次頁無

出 願 人 履 歴 情 報

識別番号 [899000057]

1. 変更年月日	1999年 9月17日
[変更理由]	新規登録
住 所	東京都千代田区九段南四丁目8番24号
氏 名	学校法人日本大学